LEVEL 3 FARM RECOVERY

Nuno Leonardo

Massachusetts Institute of Technology

for the EVB+LEVEL3+ONLINE teams

October 2003

Level 3 Farm RECOVERY

N. Leonardo

Massachusetts Institute of Technology

Abstract

This document describes the recovery mechanism developped for the Level3 online farm at CDF. A remote procedure is implemented in which a properly configured Linux rescue image executes a remotely located instruction script.

Contents

1 Recovery

Investigation and recovery from problems on individual and many nodes is achieved through extensive use of properly customized *L3 recovery floppies*.

These form convenient tools for executing required actions on nodes which fail to boot healtily. These actions can be instructed on a terminal connected to the node, or specified in advance on a file at a remote server.

Common actions include filesystem and code installation (typically necessary on a large number of nodes); hard drive inspection, partioning, filesystem creation, master boot record repairing; modifying boot loader and system configuration files . Other general, system supported, user specified actions are readily implementable as well.

1.1 Boot image

The recovery floppy is formed of a configurable boot image. This should contain a small Linux kernel with the proper drivers to boot the system in order to perform recovery operations, allowing as well customization in a simple fashion. The farm recovery procedure described here is general and can be in principle implemented using any such, rescue-oriented, available Linux image.

The particular rescue image which have been employed is the **tomsrtbt** disk image [3], described as a "floppy which has a root filesystem and is also bootable". This contains a very small distribution of Linux, with most commands useful for system recovery, most necessary drivers, and network connectivity capabilities, fitting a single floppy disk. The most attractive feature nevertheless is that the distribution comes with a number of scripts which can be used to readily create and transfer a modified image.

It contains on the other hand some objectionable features as well according to farm's requirements. It does not recognize the multi-port Ethernet cards used in Converter and Output nodes; this has required the use for recovery purposes of the on-board port. The current version does not contain either the `sfdisk` Linux command, which would have otherwise been specially useful (it would facilitate the typical process of hard-drive partitionning+installation); nor `ext3` filesystem type creation features.

Also bothersome was the necessity of explicit login (requiring direct name/password typing at the console); this was fixed by compiling an appropriately modified `sulogin.c` code (where `getpasswd()` has been removed), enforcing static linking, and transferring it to the image.

Installation

An appropriate tar ball containing the *tomsrtbt* image can be downloaded from one of the listed [3] sites. The accompanying script `install.s` allows to transfer the raw image to a floppy disk; `fdformat` should be executed on the floppy in case it is not properly formated already.

Image customization requires a few steps. Executing `unpack.s tomsrtbt.raw` creates an expanded version of the raw image structure. Files located in the directory `tomsrtbt-`*version*`.unpacked/2/` can be modified as desired. An accordingly customized image can then be produced and installed simply by executing the scripts `buildit.s` and `install.s`, respectively, from the unpacked directory.

The Level3 customization presented is general, its dependence on image releases being minimal; although, extra functionality may become available.

1.2 *Remote* procedure

The raw rescue system is not adapted to the farm imperative of simultaneous recovery of many computers. A procedure was therefore developed allowing fast recovery of a large number of nodes in the farm, in a shortest time period – crucial for having the full farm readily operational, and not disturbing data taking. Indeed, it has allowed for safe, full-scale farm shutdowns and rebootings, with the guaranty of a reasonably short time for recovery of possibly many (typical case) failing nodes.

To decrease the time of an individual recovery operation, the original login program is replaced by a modified one wherein that process is made automatic; sleeping times, reserved for other systems' or interactive usages, are reduced or simply eliminated.

To cope with the necessity of dealing effectively with many nodes, it is essential to make the recovery procedure completely automated, whith no need for user's direct input. This is achieved by adapting the recovery system's configuration files so the desired actions are performed at the very first stages of system initialisation, while other, default actions are not executed if not necessary.

A simplified initalization table (`/etc/inittab`, the configuration file to **init**) is defined, where the instruction `::sysinit:/etc/rc.S` is included. This instructs initial execution of `/etc/rc.S`, where some image configuration scripts are sourced, and at the end of which a call to a procedure execution script is *added*. The later can contain a set of intended actions on the node, ending for example with a `/sbin/reboot` instruction. The advantage of these features is the earliest execution of the recovery actions right at initialization level, and interruption of the default initialization process itself once the intended actions have been completed. Effectively, unnecessary, time consuming actions and configurations do not occur, by avoiding definition of common *run level*'s and their execution.

Furthermore, a *remote procedure* is put together where the procedure execution script is stored on a remote node, accessible via the network. This clearly favors generality and convenient procedure development. Indeed, once such an image has been created and transferred to a portable disk, the later can then be used quite generally, for execution of whatever actions are defined on a convenient script file residing on a remote, permanently accessible node.

This is achieved in practice as follows. A script arbitrarily named *fixnode.all.remote.floppy* is defined, and executed at the end of the file `/etc/rc.S`, whose function is to

1. configure the network,

2. mount an `nfs` accessible location, and

3. execute a remotely located script where user-specified actions are pre-defined.

Notice also that, if after a certain number of actions have taken place (e.g. network configuration) one wishes to still have access to a shell using a console directly connected to the node, this is made possible simply by including a call to a login program. This program can be e.g. the `sulogin` alluded to before, and can be included with the image or, e.g. for lack of available space on the floppy, simply placed on the remote location.

1.3 Recovering filesystems

1.3.1 Filesystem corruption

The recovery procedure here discribed was originally developped to cope with a persistent hardware failure affecting consistently a considerable fraction of the farm. It was verified that filesystems in those nodes were being progressively corrupted with time, eventually inducing their failure. In that case, it would then interrupt data taking, requiring an expert to arrive and remove it temporarily from the online hardware database.

No practical hardware solution to this problem was found, nevertheless its effects were surpassed through persistent maitenance efforts. These involved periodic (initially weekly) rebooting of the farm, enforcing checks of all filesystems every time, and systematic recovery of those shown defective.

Track was kept of the frequency of filesystem failures over the farm, both in a recovery, automatically generated log file and node statistics page [2]. This allowed for an assignment of a failure expectation to nodes in the farm, use of which has been made in preventive replacements and posterior hardware upgrades, as well as in progressive transfer of nodes with higher failure rates to less critical, offline usages.

1.3.2 Automating procedure

The remote recovery procedure is used for repairing damaged filesystems on the Level3 farm. An image is created containing the script *fixnode.all.remote.floppy*, which is directly executed at initilization time. Its description follows below.

configuring network

A set of `IP addresses` are *reserved* in the Level3 Ethernet network for use in the recovery process. These are explicitly mentioned in the `/etc/hosts` file of the server; e.g.

```
192.168.23.240        b0l3rec0.fnal.gov b0l3rec0
...
```

To each image floppy is assigned a distinct address among this set, which it uses for accessing the private network. The following set of instructions are sufficient for configuring a network interface.

```
ripadr=192.168.22.240
ifconfig eth0 \$ripadr
ifconfig eth0 broadcast 192.168.255.255
ifconfig eth0 netmask 255.255.0.0
route add -net 192.168.0.0
```

nfs-mounting

A remote node should be available as `nfs` server, with a directory containing the necessary materials for implementing the recovery procedure.

Any available, *failure-free* Linux box in the farm can in principle be used as server for the recovery process. The directory containing the recovery materials would have to be `nfs` exported; this is done by creating/editing the file `/etc/exports` with contents as in

```
/recovery    192.168.0.0/255.255.0.0(rw,no_root_squash)
```

(where the IP address/netmask pair is used to avoid access control problems for `nfs` related daemons), and issuing the commands `rpc.`**nfsd** and `rpc.`**mountd** (or just **exportfs -a** in case `nfs` is already up and running).

In the Level3 farm Gateway1 has been used for this purpose. Recovery materials are located in `/home/recovery` directory (where the `/home` directory is permanently `nfs` exported to the entire farm). Its *remote* sub-directory can be mounted as follows.

```
mkdir /mnt/extdisk
mount -t nfs 192.168.24.11:/home/recovery/remote /mnt/extdisk
```

write permissions are kept for log purposes only, and are otherwise not required.

execute remote script

Having now successfully mounted the remote recovery directory, a call to a main script (*fixnode.all.remote*) is performed.

```
/mnt/extdisk/fixnode.all.remote
```

This will be expected to end with a `reboot` instruction. However, if that is not the case, or if the remote script is not found at all, a call to a login command may be used to allow direct console input.

```
/mnt/extdisk/bin/sulogin
```

This ends the image's residing script, *fixnode.all.remote.floppy*.

1.3.3 Installing filesystems

Filesystem recovery involves a number of steps. These are instructed in the main remote recovery script (*fixnode.all.remote*). This script, which resides on the remote recovery directory in Gateway1, reads in filesystem and node information from accompanying configuration files, recreates the specified filesystems by extracting their standard contents from available tarballs also located remotely in the server; it ends with a node rebooting instruction, after writing some log information as record of what was done.

read in recovery information

Berore executing the procedure, information needs to be specified regarding the filesystem to be recovered. This is done in a file (*recover.config-remote*) residing remotely (Gateway1). It may contain the name of the filesystem to be recreated, that of the corresponding partition, as well as the name and origin of an appropriate tarball; e.g.

```
hda7   cdf_260.tar     192.168.23.60   /cdf
```

re-create filesystem

The specified filesystem can be created using the `mke2fs` Linux program, destroying the previously existing corrupted one.

If a full filesystem check (`fsck`) is desired each time the machine boots, this can be forced by adjusting the maximal mounts count between two checks to unit; alternatively, the maximal time between two checks can also be imposed. This is done using `tune2fs`, and should be instructed at this stage, before mounting the filesystem.

expanding contents

The intended contents of the associated partition may be transferred to the node by expanding an appropriate tarball, once both the remote `tar` file location and the local partition are mounted on the image's recovery system.

The procedure expects the filesytems' tar files to be in a pre-defined location (namely, the `tarballs` subdirectory); such tarball can be produced from an equivalent healty node as in the following example.

```
ssh root@b013260;  cd /cdf
tar -cvpf /home/recovery/tarballs/cdf_260.tar ./*
```

updating IP address

In case the root filesystem happens to be the one recovered, node unique configuration information, as its IP address, needs to be specified.

Firstly one extracts a node's hardware identification, as the Ethernet card's MAC (Media Access Control) address; this is displayed in the **ifconfig** output following the string `HWaddr`. Then the associated IP address is extracted from a tab-delimited list of hostname, IP and MAC addresses. Such a list (*ipmaclistL3-remote*) is produced readily by executing a simple script over the farm which extracts the addresses' information from ifconfig output. It should be updated whenever there are relevant hardware modifications; including e.g. node swaping and Ethernet card replacement.

The thus found node's IP address should be used to update the Ethernet port(s) configuration settings; e.g., the file `/etc/sysconfig/network-scripts/ifcfg-eth0` for the first Ethernet port.

logging

A record of the performed recovery is entered in a log file, `stat.log`. This should contain information identifying the recovered node, filesystem, and date the procedure was performed.

checking for floppy

The procedure should terminate with a reboot instruction for the node just recovered. Before this instruction can be issued the recovery floppy needs to be removed from the node's drive, avoiding repetitive execution; the procedure halts and waits for that to happen. Checks are regularly made for the presence of the floppy; additionally, a temporary file with a suggestive name is *touched* on the server. Only once it is no longer detected does the system proceed with the final **reboot** instruction.

further

The automated procedure may be interrupted at any stage (prior to the final reboot instruction) simply by issuing a call to a login program, as the remote `sulogin`, and be continued in a interactive fashion. This may be useful in cases when e.g. several filesystems are to be installed, or for post-installation inspection purposes.

Updates to the recovery procedure are implemented remotely; thus, without the need to e.g. modifying recovery images.

1.4 Installation through recovery

The Level3 farm recovery floppy can naturally be employed for the purpose of node installation; this corresponds essentially to recoverying *all* filesystems. It has been used in cases of new node, new hard drive, and new system installations.

partitioning

Disk (re-)partitioning may be necessary, and can be conveniently done using the `sfdisk` Linux partition table manipulator. First, a file with the intended partition

map should be created; this can be done by dumping the partitions of an existing, identically partitioned device, as shown below.

```
sfdisk -d /dev/hda > hda.out
```

A file is produced this way with the appropriate format to serve as input to `sfdisk` as follows.

```
/sbin/sfdisk /dev/hda < hda.out
```

The option $--force$ to `sfdisk` can be used, in which case one should be even more certain of the requested actions.

In case the hard drive to be partitioned is not (cannot be) mounted in a local root filesystem, thus rendering the `sfdisk` facility unavailable, one may use an uncustomized version of the recovery floppy (or raw *tomsrtbt*) to perform a manual partitioning using the standard `fdisk` manipulator.

multi-recovery

Once the disk has been re-partitioned as intended, the remote recovery floppy can be used to complete the installation. The associated remote script would have to perform the remaining, usual actions of filesystem creation (`mke2fs`), swap areas setting up (`mkswap`), filesystem mounting and contents transfer from tarballs over the network, MBR fixing, and reboot.

1.5 Generality

The recovery floppies execute whatever instructions are specified on a main remote script; this is named *fixnode.all.remote* and resides on the directory `/home/recovery/remote/` at the recovery `nfs` server (Gateway1).

The performed actions can be a simple call to a shell login program, filesystem recovery or installation as specified in previous sections, or any other specified action supported by the image's system (or by the node's system in a post recovery stage using the chroot environment).

Other commonly useful, available commands include:

- `fdisk`, for disk paritioning
- `chroot`, for executing system commands on the node
- `lilo`, for MBR repair; e.g. `hda1` being the root filesystem,
  `chroot /dev/hda1 /sbin/lilo`

References

[1] N. Leonardo for the EVB/L3 team, *Event Builder and Level 3 Manual for Experts*, CDF Note 6138, http://www-cdfonline.fnal.gov/evbl3shift/evbl3pager.html.

[2] N. Leonardo for the EVB/L3 team, *EVB/L3 experts online page*, http://www-cdfonline.fnal.gov/evbl3shift/evbl3pager.html
Recovery, http://www-cdfonline.fnal.gov/evbl3shift/pager/recovery/
Node statistics, http://www-cdfonline.fnal.gov/evbl3shift/pager/recovery/nodelist.html

[3] T. Oehser, *The most GNU/Linux on 1 floppy disk*, http://www.toms.net/rb/

[4] The Femi Linux page, http://www-oss.fnal.gov/projects/fermilinux/
v7.3.1 distribution, http://www-oss.fnal.gov/projects/fermilinux/731/home.html
v7.3.1 ftp, ftp://linux.fnal.gov/linux/731a/
v7.3.1 images, ftp://linux.fnal.gov/linux/731a/i386/images/

[5] The Official Red Hat Linux Customization Guide,
http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/custom-guide/index.html

[6] Fermi Workgroups,
http://www-oss.fnal.gov/projects/fermilinux/common/workgroups.maintainers.html

[7] UPS, UPD and UPP page at Fermilab,
http://www.fnal.gov/docs/products/ups/

UPD and UPD distribution server,
http://kits.fnal.gov/

Main ftp server containing available product releases,
ftp://ftp.fnal.gov/products/

A Recovery image configuration and procedure implementation

A.1 Configuration files

inittab

```
::sysinit:/etc/rc.S
::ctrlaltdel:/bin/reboot
::shutdown:/etc/rc.0
```

rc.S

```
#!/bin/sh
PATH=/bin
BT=1
#Read from image
mount -o remount /
mount -a
mount -o ro /dev/fd0u1722 /fl
. /fl/settings.s
update &
cp /fl/rc.* /fl/*.s /bin
gzip -d /bin/*.gz
. /bin/rc.custom
umount /fl
#Execute local script
/fixnode.all.remote.floppy
```

Note: When a PC is booted the BIOS runs its power-on-self-test (post) and then looks for more boot information in the boot sector of the available device: (i) typically in the Master Boot Record (MBR) of the hard drive, (ii) in floppy if present (and booting device order set accordingly in BIOS). The Linux kernel then loads **init**, the general process dispatcher, using `/etc/inittab` as its configuration file, where run levels are defined and the default is specified. The initialisation table consists of records of the format: `id:runlevels:action:process`. *Usually*, the first script run by init is `rc.S`, then it drops down and runs the default run level (usually 2).

In the present recovery implementation the procedure is performed just *before* run levels are executed.

A.2 Include a new binary

Below is shown how to include a new binary file as part of a customized image. The case of *sulogin* (single user login) program is provided in detail as an example.

```
Download source code; e.g. search as in:
    http://rpmfind.net/linux/rpm2html/search.php?query=SysVinit

Unpack
    > rpm -ihv SysVinit-2.78-5.src.rpm

Modify code; in sulogin.c comment out getpasswd() call
    /*
    while(1) {
        if ((p = getpasswd()) == NULL) break;
        if (pwd->pw\_passwd[0] == 0 ||
        strcmp(crypt(p, pwd->pw\_passwd), pwd->pw\_passwd) == 0)
    */
        sushell(pwd);
    /*
        printf("Login incorrect.\\n");
    }
    */

Remove other un-necessary features; e.g. PCMCIA related

Compile and enforce static link
    > cc -Wall -O2 -D\_GNU\_SOURCE   -c -o sulogin.o sulogin.c
    > cc -s -static -o sulogin sulogin.o -lcrypt

Transfer executable to unpacked image directory
   > cp sulogin  tomsrtbt-[version]/tomsrtbt-[version].unpacked/2/sbin/

Call program; impose sulogin at specified run level
   > vi tomsrtbt-[version].unpacked/2/etc/inittab

      id:4:initdefault:
      si:S:sysinit:/etc/rc.S
      rc:5:wait:/etc/rc.M
      ca::ctrlaltdel:/bin/shutdown -t5 -rfn now "CtlAltDel"
      l0:0:wait:/etc/rc.0
      l6:6:wait:/etc/rc.6
      ~~:S:wait:/sbin/sulogin
      c1:5:wait:/sbin/sulogin
```

A.3 Create a recovery floppy

```
Go to Tom's Root Boot Linux page:
   http://www.toms.net/rb/

Download the software from one of the listed sites, e.g.
   ftp://www.tux.org/pub/distributions/tinylinux/tomsrtbt/tomsrtbt-[version].tar.gz

Expand the .tar.gz archive:
  > tar xvzf tomsrtbt-[version].tar.gz
```

```
A) Create raw floppy:
  > cd tomsrtbt-[version]
  > ./install

B) Create a customized floppy (e.g., include myscript):
  > cd tomsrtbt-[version]
  > ./unpack.s tomsrtbt.raw
  > cd tomsrtbt-[version].unpacked/
  > cp /mylocation/myscript 2/
  > chmod +x 2/myscript
  > ./buildit.s
  > cd tomsrtbt-[version+1]
  > ./install
```

A.4 Remote recovery procedure

The common procedure for restoring a filesystem on a Level3 node involve the following steps.

1. Go to remote recovery directory; `Gateway> cd /home/recovery/remote`
2. Specify filesystem information: check/edit file `recover.config-remote`
3. Check for existence of corresponding tar file in `tarballs` directory; otherwise create it
4. For Output and Converter nodes, re-seat Ethernet cable (A) on on-board Ethernet port
5. Check/edit recovery script if needed, `fixnode.all.remote`
6. Execute recovery script on broken node; insert remote recovery disk on floppy drive of broken node and hit reset; this will execute locally the remote script named `fixnode.all.remote` located on remote directory
7. Remove floppy when done; wait $\sim 4min$ or check `stat.log`; the procedure will wait until the floppy has been removed from the node's drive; for Output or Converter nodes, the Ethernet cable should be re-seat into its original port
8. Check that node is up and well, starting with `ping, ssh, df`; execute specific actions known to be required by certain classes of nodes
9. Distribute relay/level3 code in case the relevant filesystem (`/cdf`) has been restored (see [2] for instructions)

Details and updates are found in the recovery page in [2].

A.5 Filesystem recovery scripts

File: fixnode.all.remote
Description: main recovery script

```
#!/bin/sh

#Get MAC address of current node
mac=‘/usr/bin/ifconfig eth0 | grep HWaddr | cut  -d " " -f 11‘
echo "The MAC address found is $mac"

#Extract node name + ip from associated $mac in ipmaclistL3-remote table
name=‘/usr/bin/grep $mac /mnt/extdisk/remote/ipmaclistL3-remote|cut -f 1 -d " "‘
ipadr=‘/usr/bin/grep $mac /mnt/extdisk/remote/ipmaclistL3-remote|cut -f 2 -d " "‘
echo "The node name found is $name with ip address $ipadr"

#Read recovery configuration: select filesystem
fsname=‘grep -v "#" /mnt/extdisk/remote/recover.config-remote|cut -f 1‘
fstar=‘grep -v "#" /mnt/extdisk/remote/recover.config-remote|cut -f 2‘
fsip=‘grep -v "#" /mnt/extdisk/remote/recover.config-remote|cut -f 3‘
fsid=‘grep -v "#" /mnt/extdisk/remote/recover.config-remote|cut -f 4‘

#Recreating filesystem on /dev/$fsname
/usr/bin/mke2fs /dev/$fsname
#Setting max mount count to 1
/usr/bin/tune2fs -c 1 /dev/$fsname

#Mount /dev/$fsname
mkdir /mnt/$fsname
mount -t ext2 /dev/$fsname /mnt/$fsname

#Expand tarball
cd /mnt/$fsname
tar xvf /mnt/extdisk/tarballs/$fstar

#Assign correct ip-address to eth0: $fsip->$ipadr
if [ $fsid = "/" ]
then
cd /mnt/$fsname/etc/sysconfig/network-scripts
sed -e s/$fsip/$ipadr/ ifcfg-eth0 > tempo
mv tempo ifcfg-eth0
fi

#Add log info to b0l3serv:/home/recovery/stat.log
echo "Rebooted node $name; recovered $fsname aka $fsid; \
      on " ‘/usr/bin/date‘ >> /mnt/extdisk/stat.log

#Wait until recovery disk is removed
mkdir /mnt/floppy
mount -t minix /dev/fd0u1722 /mnt/floppy
```

```
flp=‘df | grep floppy | wc | cut -f 1‘
umount /mnt/floppy
while [ $flp -gt 0  ]; do
 sleep 5
 mount -t minix /dev/fd0u1722 /mnt/floppy
 flp=‘df | grep floppy | wc | cut -f 1‘
 echo "Still waiting for floppy to be removed... $flp : not zero"
 umount /mnt/floppy
 touch /mnt/extdisk/remove_floppy_to_reboot_$name
done
rm  /mnt/extdisk/remove_floppy_to_reboot_$name

#Reboot the node
sleep 2
reboot
```

Auxiliary files

The remote recovery directory is `Gateway1:/home/recovery/remote/`. A description of the main remote files follows.

- `recover.config-remote.` This file contains a map which specifies which filesystem is to be recovered, and tarball information; it is read by the main script; needs to be edited before it is executed (see header for editing instructions).

  Format:

  ```
  #fs     tarball         taredFromIP     partition
  hda1    rootfs_274.tar  192.168.23.74   /
  #hda6   usr_273.tar     192.168.23.73   /usr
  #hda7   cdf_260.tar     192.168.23.60   /cdf
  ```

- `ipmaclistL3-remote.` This file contains a list of names, IP and MAC addresses of each single node; it is read by the main script; ought to always be up-to-date.

  Format:

  ```
  b0l3c01 192.168.21.1 00:00:D1:1C:6D:CF
  ...
  b0l3001 192.168.22.1 00:E0:81:04:3E:F7
  ...
  b0l3u01 192.168.26.1 00:D0:B7:A7:95:9E
  ...
  ```

  This needs to be updated whenever an hardware alteration takes place in the farm, including: installation of a new node; node swapping; Ethernet card replacement or exchange. Output and Converter nodes use a multi-port Ethernet card which is not supported by the system on the floppy; therefore one needs to

use instead the non-default (motherboard embedded) port, and it is the corresponding MAC addresses that are necessary. An updated version of the file may be readily obtained by executing a simple script as below in each node.

```
#!/bin/bash
echo "usage: getmacip_node.sh"
mac=‘/sbin/ifconfig eth0 | grep HWaddr | cut -f 11 -d " "‘
ip=‘/sbin/ifconfig eth0 | grep inet | cut -f 12 -d " " | sed -e s/addr:/""/‘
echo "‘hostname‘ $ip $mac"
exit
```

For example, for processor nodes,

```
ssh root@b0l3pcom1
touch /home/l3proxy/recovery/newlist.txt
expect /root/scripts/allProc.exp
/home/recovery/remote/scripts/getmac_node.sh \
        >> /home/l3proxy/recovery/newlist.txt
```

and similarly for Converter and Output nodes after Etherent cable re-seating.

- `stat.log`. This is the general recovery log file; it is automatically edited each time the procedure is used, with information about the node and filesystems just recovered.

  Format:

  ```
  Rebooted node b0l3203; recovered hda6 aka /; on  Tue Sep 30 14:54:11 2003
  ```

- `tarballs/`. This directory should contain (at the time of execution) the appropriate tar file with the filesystem contents to be installed; correct description of these files should appear on the file "recover.config-remote"; appropriate tarballs may be available on: Gateway2:tarballs/tarballs.reservoir

- `bin/`. This directory contains binaries to be executed by the recovery system; e.g. `sulogin`.

- `scripts/`. This directory contains various scripts pertaining to the procedure.

- `store/`. This is a backup directory; prior to modifying any of the files one ought to be sure that a backup version is saved.